Microsoft Exchange Server

Exchange 2000 Front-end / Back-end Topology

White Paper

Published: July 2000
Author: Sean Lyndersay, Program Manager, Microsoft Exchange Server

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2000 Microsoft Corporation. All rights reserved.

Microsoft, Windows, Active Directory, Outlook, Jscript, and IntelliMirror are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners. The example companies, organizations, products, people and events depicted herein are fictitious. No association with any real company, organization, product, person, or event is intended or should be inferred.

Table of Contents

Introduction....1

Multi-server Environments....2

How Front-end / Back-end Topology Works....3
- HTTP (Outlook Web Access and Web Folders)....3
- POP3 and IMAP4....8

Deployment Considerations....9
- Recommended server configurations and ratios....9
- Scenarios....9
- Intranet Scenarios....10
- Extranet Scenarios....12

Configuration....14
- Configuring a Front-end Server....14
- Configuring a Back-end....16
- Configuring Firewalls....20
- Additional Configuration....22
- Front-end/Back-end Topology: Checklist....24
- Front-end/Back-end Topology: Troubleshooting steps....24

For more information....25

Exchange 2000 Front-end / Back-end Topology

White Paper

Published: July 2000

For the latest information, please see http://www.microsoft.com/exchange/

Introduction

Microsoft® Exchange 2000 Server supports the deployment of Exchange in a manner that distributes server tasks between multiple computers, namely front-end and back-end servers. Generally speaking, a front-end server accepts requests from clients, and proxies them to the appropriate back-end server for handling. This document discusses this general architecture with particular attention paid to the HTTP protocol.

Assumed Knowledge

This document assumes that the reader understands Outlook Web Access (HTTP), POP3 or IMAP4 in a standard Exchange deployment, and, in addition, that basic Exchange 2000 and Microsoft Windows® 2000 Internet Information Services (IIS) concepts are understood. For more information on Exchange and IIS, please see the forth-coming document: “Exchange 2000 and Windows 2000 Internet Information Services: Integration and Co-existence”

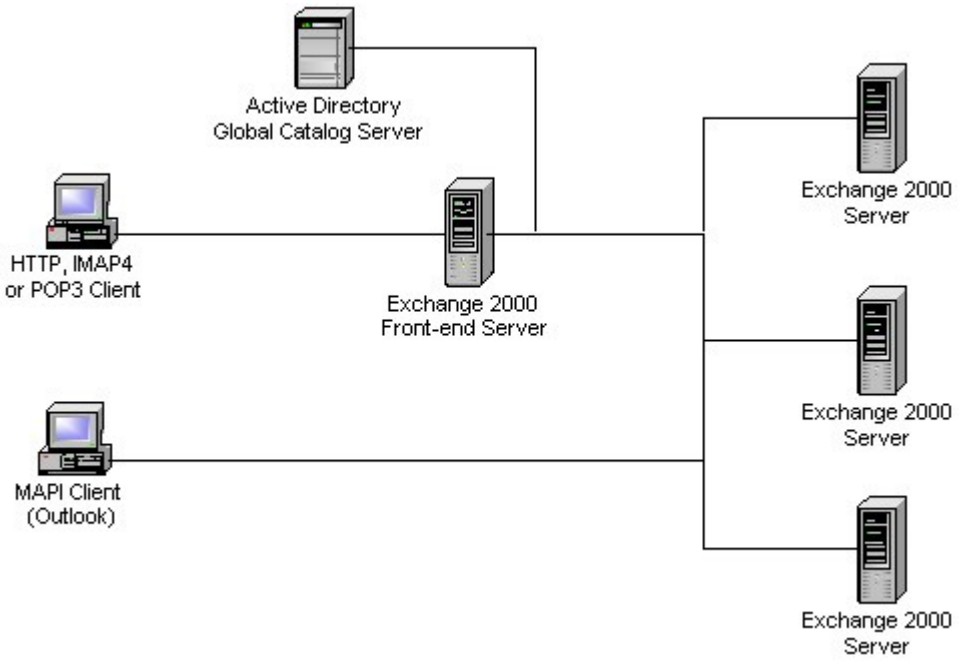

Figure 1: A Simple Exchange 2000 Front-end/Back-end Topology

Multi-server Environments

Microsoft recommends the front-end and back-end server topology for multi-server organizations using Outlook Web Access (HTTP), POP3 or IMAP4, and for organizations wishing to provide HTTP, POP3 or IMAP4 access to their employees over the Internet. Upon receiving a request, the front-end server uses the Lightweight Directory Access Protocol (LDAP) to query the Windows 2000 Active Directory and determine which back-end server holds the requested resource.

Note A front-end server is a specially configured Exchange 2000 server. A back-end server, however, is just a normal Exchange 2000 server. There is no configuration option to designate a server as back-end. The term “back-end server” is used to refer to all servers in an organization that are not front-end servers, once a front-end server is introduced into the organization. Without a front-end, the term is meaningless.

Using a front-end/back-end deployment has the following advantages:

- **Single namespace.** The primary advantage of a front-end and back-end server architecture is the ability to expose single consistent namespace. You can define a single namespace for users to access their mailboxes (for example http://mail for Outlook Web Access). Without a front-end, each user must know the name of the server that is storing their mailbox. This complicates administration and compromises flexibility, because every time your organization grows or changes, requiring you to move some or all mailboxes to another server, you must inform the users. With a single namespace, users can use the same URL or POP3 and IMAP4 client configuration, even if servers are added and removed, or if mailboxes are moved from server to server. In addition, creating a single namespace ensures that Outlook Web Access, POP3 or IMAP4 access remains scalable as your organization grows.

- **Offload Processing.** Exchange 2000 servers can be configured to support SSL (Secure Sockets Layer) traffic between the client and the server, to protect the traffic from third-party interception. However, encrypting and decrypting message traffic uses processor time. Therefore, front-end and back-end server architecture provides an additional advantage when using SSL encryption, because the front-end servers can handle all encryption and decryption processing, which improves performance by removing processing tasks from back-end servers, while still allowing the data to be encrypted between the client and the Exchange servers. In addition, HTTP compression can be offloaded to the front-end processors, allowing clients in low-bandwidth scenarios to retrieve data much more quickly and improving network availability in the general case.

- **Firewalls.** The front-end server can be positioned as the single point of access on or behind an Internet firewall, which is configured to allow only traffic to the front-end from the Internet. Because the front-end server has no user information on it, it provides an additional layer of security for the corporation. In addition, because the front-end server can be configured to authenticate requests before proxying them, the backend servers are protected from Denial-of-Service attacks.

- **Increased IMAP4 access to Public Folders.** The IMAP4 protocol allows a server to refer a client to another server. Exchange 2000

servers support this functionality in cases where a public folder store on a particular server does not contain the content requested. However, this requires a client that supports IMAP referrals. Most existing clients do not support them. When a non-referral enabled IMAP4 client (see RFC 2221 and RFC 2193) connects through a front-end server, the client has access to the entire public folder hierarchy. When a front-end server proxies a command to a back-end server, it will automatically automatically handle any referral response that is passed back when accessing a folder that is not available on the back-end server, making the referral transparent to the client

The front-end / back-end architecture supports the HTTP, POP3, and IMAP4 protocols. SMTP and NNTP (file system only, because no store is on the server) protocols may be installed on the front-end server, if desired. The MAPI protocol (used, for example, by Outlook 2000) is not supported by the front-end / back-end architecture. The advantages of the front-end / back-end topology are not useful to MAPI clients for two reasons. First, MAPI clients have built-in support for handling cases when users are moved from one server to another, or when content is not available on a server. Second, MAPI is not an Internet protocol, so the Internet availability features of the front-end do not affect it.

How Front-end / Back-end Topology Works

While the general functionality of the front-end server is to proxy requests to the correct back-end servers on behalf of the clients, the exact functionality of the front-end is protocol dependent. Each of the supported protocols will be discussed separately.

HTTP (Outlook Web Access and Web Folders)

Whether generated by a browser, or a specialized client like the Windows 2000 Web Folders feature, HTTP requests from the client computer are sent to the front-end server. The server uses Active Directory to determine the back-end to which the request should be proxied. The precise mechanics of the determination vary depending on whether user mailboxes or public folders are being accessed, but the effect is that the data being requested is found wherever it may be in the Exchange organization, assuming appropriate permissions have been granted to the user accessing the data

Once the correct back-end server is determined, the front-end forwards a request to the back-end. The request is identical to the original request sent from the client. In particular, the HTTP host header (which is generated by the client, and matches the name of the front-end server to which the request was sent) remains unchanged, even though the new request is not sent to a server with that name. The host header setting ensures that the request is handled by the Exchange virtual server, which must be configured on each back-end server to handle front-end server requests.

The front-end will always contact the back-end over TCP port 80 (the default HTTP port), regardless of the port the client used to contact it. This means first, that SSL encryption will never be used between the front and back end servers, though the client in communication with the front-end may use it. Second, HTTP virtual servers that differentiate themselves from others only by port number are not supported in a front-end/back-end topology. For instance, if a particular back-

end server had an HTTP virtual server listening at port 8080, it would be accessible only by pointing the client directly to the back-end server (e.g. http://backend:8080/data). It would not be accessible through the front-end.

The request from the front-end server is processed normally on the back-end, and the response is sent unchanged through the front-end server back to the client. In most cases, the back-end considers the front-end to be simply another HTTP client.

The client, therefore, never realizes that the request was not handled on the front-end server.

Finding User Mailboxes

To provide access to Private Folders through HTTP, a virtual directory must be configured which points to the private folders. When setting up this virtual directory through the Exchange System Manager (ESM), the SMTP domain name can be selected. A virtual directory must exist for each SMTP domain name that is to be accessed through the server. The default virtual directory, named "Exchange", is created when Exchange 2000 is installed and points to the default SMTP domain name for the organization.

When the front-end server detects that a request is being made to a location within the private folders (based on the virtual directory or virtual server root directory being accessed), it contacts an Active Directory Global Catalog server in the domain and determines which back-end Exchange server has the mailbox for the user. The front-end server executes the query against Active Directory by using LDAP.

Details

There are two ways for a user to connect Outlook Web Access over HTTP. The first is known as implicit logon, and is provided as a shortcut. Implicit logon is only used for the first access to a mailbox in a session, and only when the username is omitted from the first request (for example, http://server/exchange/). For it to work, authentication must be enabled on the front-end server. Once the user is authenticated, the authentication information is used to look up the mailbox associated with the user in Active Directory, and the back-end server on which the mailbox is located. The URL is then updated with the username and sent to the correct back-end server. Implicit logon is only useful for using Outlook Web Access – specialized HTTP clients generally will not use this technique.

Explicit logon includes the username in the URL (for example, http://server/exchange/username/). This form must be used when the front-end is not configured to authenticate users (see the Authentication Issues section below for more details), or when a user is attempting to access a mailbox that is not his own (but to which he has access). When the front-end server receives a request in this form from a client, the username is extracted from the URL and combined with the SMTP domain name associated with the virtual directory or virtual server (this association is made when the virtual directory or virtual server is created) to construct a fully qualified SMTP address. The front-end server then looks up this address in Active Directory, and determines which back-end server has the mailbox associated with the address. The request is then forwarded to that back-end server which will process the request as though it was directly from the client. The response it generates is returned to the front-end server, which returns it unchanged to the client.

Finding Public Folders

Just as virtual directories must be configured for private folders, virtual directories must be created for each of the public folder trees that are to be made accessible over HTTP via the front-end. During installation of Exchange 2000, a virtual directory, “public,” is created to allow access to the default (MAPI-accessible) public folder tree. When other public folder trees (to host applications) are created, virtual directories must be created using the ESM, to expose these trees via HTTP. Identical virtual directories must exist on the front-end and on all back-ends that host the public folder tree.

A request made to a URL within a public folder tree is handled slightly differently when accessing the default (or MAPI-accessible) public folder tree than when accessing application public folder trees (also known as application Top-Level Hierarchies (TLH), or non-MAPI TLHs).

In both cases, however, the goal is twofold: availability and consistency. First, if the data being accessed is available in an Exchange 2000 public folder somewhere in the Exchange organization, and is accessible via HTTP, then it will be provided to the user. Second, as long as the server is available, the same public folder server will always be used to service every request. This latter feature ensures that the user will see the same data each time he accesses the public folder trees through the front-end (including message read/unread status, which is stored on individual servers, and is not replicated between public folder servers). The fact that users will always reach the same back-end server is also important for server-based applications that maintain session state, such as some built with Active Server Pages.

Details

When accessing the default public folder tree, an attempt is made to maintain parity with MAPI clients, such as Outlook 2000. When administrators configure a user’s profile, the user’s mailbox store is associated with a particular public folder store somewhere in the organization (sometimes on the same computer as the private store, but sometimes on dedicated public folder servers). This is the public folder store that is displayed in the Outlook 2000 client. The front-end server authenticates the user extracts this information from Active Directory, and forwards requests to the authenticated user’s public folder server. This mechanism ensures that users switching between Outlook 2000 and Outlook Web Access will see the same data, and takes advantage of the load-balancing that the administrators have configured into the system. If the authenticated user does not have a mailbox associated with it (for instance, when giving Windows NT 4.0 users access to Exchange 2000 mailboxes), the front-end server will treat accesses to the default public folder tree as it does access to the application public folder trees.

Whereas default public folder tree servers have an association with mailbox stores due to their MAPI heritage, application public folder servers do not have such an association. Thus, when a client attempts to access application public folder trees, the front-end server contacts the Active Directory Server to find a list of all Exchange servers in the organization that have a replica of the particular application public folder tree that the client is attempting to access. This list is filtered to remove any Exchange 5.5 servers in the Organization, because they do not support the HTTP extensions (WebDAV) supported by Exchange 2000. The

user's authentication token is used to hash† over the list of servers, ensuring that users are load-balanced across the available servers, while at the same time ensuring that a particular user's requests are always processed by the same back-end server, regardless of the HTTP client used.

Because public folder replication is configured on a per-folder basis, the presence of a public folder tree hierarchy on a back-end server does not necessarily imply that the server will have a replica of the contents of a particular folder. This level of detail is not maintained in Active Directory, but instead is maintained as a property on each folder in the public folder store. Therefore, special handling is required when the server selected by the front-end, using the procedure in the previous paragraph, does not contain a replica of the specific data requested by the client. When a front-end server sends such a request, the back-end server returns a special response (this is the only case in which the back-end will treat requests from a front-end differently from those directly from clients). The back-end's response contains the list of servers that actually have the content of that particular folder. The front-end will not pass this information back to the client. Instead, it executes the same hashing technique on the new list of servers – again, to ensure load-balancing. Thus, in cases where partial replicas of public folder trees are used, the front-end may have to perform two HTTP requests to satisfy the client's single request. However, in processing the client's request, the front-end will remember which servers had the content, allowing it to avoid extra requests when data in the same folder is accessed in the future.

The front-end server maintains caches that substantially reduce the number of queries sent to Active Directory and back-end servers for both public and private folder accesses. Cache information expires after a period of time, and is also reset when changes in server configuration are detected.

Two other issues are worth discussing. First, if a back-end server is down for maintenance or is otherwise inaccessible via HTTP, the front-end server will be unable to connect to it. The front-end server marks that server as "unavailable" for a period of ten minutes, and sends the request to a different server, if there are any available (the request will fail if there are none available). As long as the server is marked in this way, requests that would otherwise be sent to that server are automatically directed to other servers. Therefore, after a back-end server is returned to production, it may be inaccessible through the front-end for as long as ten minutes, because the front-end may continue to think that the back-end server is unavailable.

The second issue is that if the list of available servers is permanently changed for any reason, it may cause all users of application public folder trees to be directed to different servers. Users of the default public folder tree (which is the one displayed by the folder view in Outlook Web Access) will not experience any changes in their access due to changes in the list of available servers. A permanent change includes addition or removal of servers from the list of servers that host a particular public folder tree hierarchy. The nature of the server selection algorithm means that any changes to this list affect where all users are directed. For example, suppose that two back-end servers, numbered #1 and #2, with a particular application public folder tree are deployed. Then, given access by six different users, A, B C, D, E, and F, they would be spread out over the two

† A hashing algorithm applies a given number (in this case, the user's security token), and uses it to generate a position in a list, such that the distribution of all possible inputs is even over the list.

servers as follows: users A, B, and C would get their data from server #1 and users D, E, F would get their data from server #2 (remember that this load-balancing is done transparently – users will not know which back-end server is actually handling the request). Then, another server is added, server #3. Now the users will be spread out as follows: users A and B on server #1, users C and D on #2 and users E and F on #3. So users A, B and D didn't change servers, but users C, E and F did. This even load-balancing is the goal behind the server selection algorithm (the "hashing") but it has the side effect that the spread of users across servers is dependent on the number of servers. For a particular user, if the server processing their requests changes then their read/unread state will be reset. Users of web-based applications (running in application public folder trees), which maintain session state, may need to restart their application session if they were using the application during the transition period. Typically a user will not be able to tell that that anything physical has changed due to the transparent nature of the front-end server, except by observing that the read/unread state has reset or that their application is exhibiting problems. Therefore, it is recommended that the administrators inform their users before adding or removing application public folder tree servers. This only affects users of application public folder trees.

Authentication Issues

The front-end server can do one of two things. It can either authenticate the user itself, which yields certain advantages, or it can proxy the authentication straight through to the back-end server. It is important to note that, either way, the back-end server will also perform authentication. Microsoft recommends that the front-end server be configured to authenticate users, particularly when exposed to the Internet.

Only HTTP 1.1 Basic authentication is supported in a front-end/back-end scenario. Basic authentication is a fairly simple authentication mechanism defined by the HTTP specification that lightly encrypts the user's username and password before sending it to the server. To achieve real password security in a front-end/back-end topology, SSL encryption should be used in combination with Basic authentication.

Neither Windows Integrated Security, the package that supports both NTLM and Kerberos authentication, nor HTTP 1.1's Digest authentication, are supported by Exchange 2000 HTTP front-end servers.

Basic authentication does not support single sign-on. Single sign-on refers to the scenario where users logs in once to a Windows-based computer, authenticating against a domain, and then can access all resources and applications without ever being forced to re-enter their credentials. Internet Explorer 4.0 and later do enable this functionality for web applications, including Outlook Web Access if, and only if, the server being accessed has Windows Integrated Authentication enabled. Because front-end servers do not support this authentication mechanism, users must always re-enter their credentials when accessing HTTP applications, users will always be prompted for authentication, even if they are already using Windows to log on.

When authenticating against a front-end, the username must be entered in the format *domain**username*.

Pass-through Authentication

In a pass-through Authentication scenario, the front-end is configured with Anonymous authentication, so it does not challenge the user for an Authorization header. However, the back-end will challenge the user, and the challenge and its response will be routed unchanged through the front-end. Pass-through authentication should not be used, unless it is not possible for the front-end to authenticate (for instance, in a locked down DMZ. See the Scenarios section below).

Dual Authentication

In a dual authentication scenario, both the front-end and the back-end are configured to authenticate the user. While this does have the disadvantage that both servers must perform authentication, it is not as bad as one might suspect. With HTTP Basic authentication, the authentication information is carried in every request sent from the client to the server, and is passed on to the back-end by the front-end. The authentication information in the request will suffice for both servers, so after the front-end has requested authentication information from the user, the back-end will receive the same information and need not re-request it from the user. Microsoft recommends that front-end servers be configured to perform authentication whenever possible. If it is not possible to enable authentication, then implicit logon will not work (though access will still be available using explicit logon) and it will not be possible to load balance Public folder requests.

POP3 and IMAP4

When you use a front-end server, the names of the actual servers that are hosting the mailboxes are abstracted from the users. POP3 and IMAP4 client configuration is simplified and easily standardized, since every user's client is configured to connect to the one host name shared by the front-end servers. Moves of users between servers are transparent to the user and require no client reconfiguration. SMTP may be installed on the front-end server, or a separate SMTP server may be set up, to enable mail submission.

Details

A POP3 or IMAP4 client will issue a login request that contains the name of the mailbox to be accessed to the front-end server. The front-end server does not authenticate the user. It simply uses Active Directory to determine which back-end to proxy commands to. The login request is then proxied from the front-end server to the appropriate back-end server, where authentication is performed. The back-end then sends the results of the login operation back to the front-end server, which then returns the results of the operation back to the client. Subsequent POP3 or IMAP4 commands are handled in a similar fashion. Mail submission via SMTP on the front-end works in exactly the same way it does on any other Exchange 2000 server.

IMAP4 access to Public Folders

When a non-referral enabled IMAP4 client connects to a back-end server, it can only access public folders that have a replica on the user's home server. Public folders with replicas on other servers are available to IMAP4 clients, but access to these folders requires the user of a referral-enabled client.

Details

A referral-enabled client issues special commands to an IMAP4 server to determine the entire list of public folders available to it. When selecting a public folder that does not have a local replica, the server responds to the client's request with a referral URL that contains the name of the server that has the data. The referral-enabled client will then create a new connection to that server, to fetch the appropriate information.

When using an IMAP4 client against a front-end server, the front-end server acts as a referral enabled client. It transparently maps regular client requests to their referral counter parts, making the entire list of public folders available to a non-referral enabled client. When the front-end server receives a referral response from the back-end server, it does not pass this response back to the client. Instead it follows the referral for the client and makes a connection to the appropriate back-end server that has the data. The back-end server then responds with the requested item, which the front-end then relays back to the client.

Deployment Considerations

Recommended server configurations and ratios

Server configuration is dependent on a large number of factors, including the number of users per back-end server, the protocols used, and the expected use. The configuration of particular models should be done in consultation with a hardware vendor or consultant. More information on server sizing will be available by product launch.

As a general rule, one front-end server is reasonable for every four back-end servers. However, this number is provided *only* to give some idea of the suggested ratio, and not as hard-and-fast rule. Use this information only as a starting point. Servers need not have large or particularly fast disk storage, but should be weighted towards fast CPU and larger memories.

Scenarios

This section discusses some of the common scenarios where the Exchange 2000 front-end/back-end topology is deployed. The scenarios can be broadly divided into Intranet and Extranet scenarios, with the former focused on performance and scalability, and the latter focused on security.

For each scenario, the following topics will be discussed:

The scenario: What is the scenario, and when does it apply?

Setup Instructions: How to set up the scenario, described in general terms. Specific configuration instructions follow this section.

Discussion: What is special about this scenario? How does it work? What additional information is needed to make decisions about this scenario?

Issues: Caveats or limitations of this scenario.

Intranet Scenarios

Default Front-end/Back-end Topology

The scenario

A company wants to maintain a single namespace for their mail servers, yet cannot fit all of their users on a single server.

Setup Instructions

Set up a standard collection of Exchange 2000 servers. Set up a single Exchange 2000 server, configured as a front-end. Direct HTTP, POP3 and IMAP4 users to this server, *not* to their back-end servers. All virtual directories and servers must be configured identically on each of the servers. See Figure 1 for a diagram of this topology.

Discussion

This is the default configuration. Other than the standard front-end and back-end configuration steps, no other steps need to be taken.

Issues

It is important to note that, as long as the network permits connections between the client and the back-end servers, there is nothing preventing users from circumventing the front-end and connecting directly to the back-end. If this is undesirable, then the network routing configuration, or the back-end server configuration, must be changed to prevent direct connections between the clients and the back-ends.

Adding a Server

The scenario

A company has all their users on a single Exchange 2000 server. They want to add another computer to the organization to better manage the load. However, they don’t want anything to change for their users (specifically with regard to how they gain access to their mailboxes).

Setup Instructions

Single Exchange 2000 server, server A, makes up the original environment. Second Exchange server, server B, is added to the scenario. Both servers have users and/or public folders.

Discussion

This is supported, but is not strictly a front-end/back-end scenario. Exchange 2000 will *redirect* HTTP requests to both public and private folders if they are not located on the server the request was sent to, as long as the folders do exist somewhere in the Exchange organization. Note that this does not maintain the single namespace illusion. This configuration will return a real HTTP redirect or IMAP4 referral directly to the client, so the client application must resend the request to the new location. Most browsers follow HTTP redirects automatically. As a result, while Outlook Web Access (and other browser-based applications) will work without any extra work from the user, Web Folders or other applications

must change the URLs they use, unless they support HTTP redirects. In addition, a login referral enabled IMAP4 client is required for user's whose private mailboxes are located on Server B, and a mailbox referral enabled client is required to access public folders where a replica of that folder does not exist on the user's home server.

Single sign-on: One advantage of this technique is that it allows use of authentication methods (in particular, Windows Integrated Authentication) other than Basic, therefore enabling single sign-on functionality. In deployments where a single sign-on is important, this topology can be deployed. It lacks many of the advantages of the front-end/back-end topology and should only be deployed when the differences are well understood.

To effectively deploy this topology in a pseudo-front-end fashion, one server (or a group of servers) should be dedicated as the first server to service the request. This server should have no users on it, and no data in the public folders should be replicated to it. However, both the public and private stores should remain on the server. This server will then always generate redirects to the correct servers.

Issues

Redirect of additional Exchange virtual servers (for example, virtual servers other than the default) is not supported. The virtual directories on the two servers must be configured identically in order for the redirect to be successful.

Web Farm

The scenario

A corporation is rolling out Outlook Web Access to 200,000 users. The goal is to have a single namespace (say, http://mail) where users can go to reach their mailboxes. In addition, for performance reasons, they do not want to have a bottleneck at the front-end server, or a single point-of-failure, so they want to spread the load out over multiple front-end servers by using the Windows Network Load Balancing Service (NLBS). Other load-balancing mechanisms include DNS round-robin, or a hardware load-balancing solution.

Setup Instructions

Set up a group of servers as back-end servers in the same domain, with users distributed over the servers. Set up another group servers as front-ends and configure Windows 2000 Network Load Balancing Service (formerly known as Windows Load Balancing Service) on all these servers.

Discussion

See the Windows Help documentation on Network Load Balancing for details on how to set it up. Configuration does not require any special steps for Exchange 2000 servers.

Issues

Whatever load-balancing solution you use should ensure that each user is always sent to the same front-end server for the duration of a session, to leverage the caching and connection state information already maintained on the front-end. Windows 2000 NLBS has this ability, as do many hardware solutions.

Extranet Scenarios

Since these scenarios all require some sort of firewall, a brief discussion of the firewall requirements and recommendations follows.

Firewall Requirements

Firewalls can be provided by both software and hardware solutions. Software solutions include Microsoft Proxy Server version 2.0.

The minimum requirement of the firewall protecting the server from the Internet is port filtering. Port filtering restricts the type of network traffic through the firewall to by only allowing information sent to specific ports through the firewall.

To improve the strength of the firewall, it should also support IP filtering. This will enable restricting traffic through the firewall to only the front-end server.

Front-end behind the Firewall

The scenario

In order to achieve security, but provide access to Outlook Web Access, POP3 or IMAP4 from the Internet, a corporation wants to place the Exchange system behind the corporate firewall.

Setup Instructions

Setup an Exchange front-end/back-end environment within the corporation as normal. Configure a firewall between the front-end server and the Internet. See the configuration section below for information on how to configure an Internet firewall for use with an Exchange 2000 front-end server.

Discussion

Because the entire configuration is inside the firewall, nothing special need be done to the Exchange configuration. Once the request makes it through to the front-end server, it returns a response without any configuration changes.

IP address filtering is highly recommended in this scenario to limit requests through the firewall to only the Exchange 2000 front-end server (or servers), and block requests through the firewall to other servers in the organization.

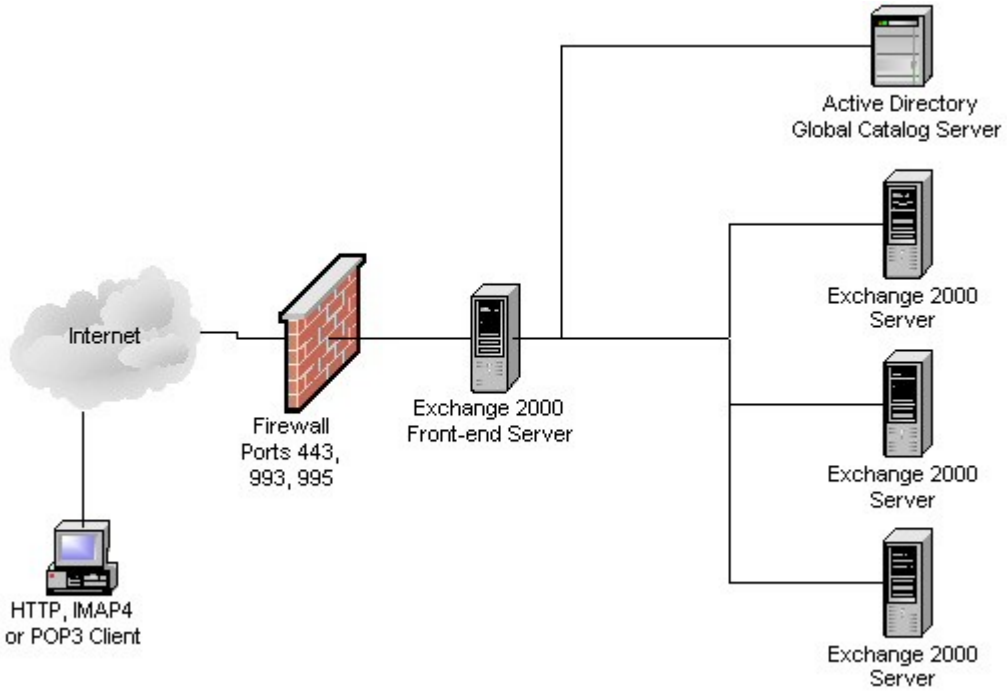

Figure 2: A Simple Exchange 2000 Firewall Topology

Demilitarized Zone (DMZ)

The scenario

In this scenario, a corporation places the front-end server between two separated firewalls. The first firewall separates the front-end from the external Internet, and only allows requests to that front-end. The second firewall separates the front-end from the internal corporation. The systems between the two firewalls lie in what is known as a Demilitarized Zone (DMZ). This configuration provides an extra measure of security by ensuring that even if the front-end server is compromised, the intruder is still isolated from the rest of the corporation, giving the corporate security team time to detect and neutralize the intrusion.

Setup Instructions

The outer, or Internet, firewall needs to be set up as normal for a firewall in this environment – limiting access only over the ports required and only to the designated front-end server. See the Configuration section for information on how to configure an Internet firewall.

The inner, or Intranet, firewall only needs to have additional ports open to support authentication and Active Directory access. The exact list depends on the balance of security and features that each corporation chooses. See the Configuration section for information on how to configure the Intranet firewall.

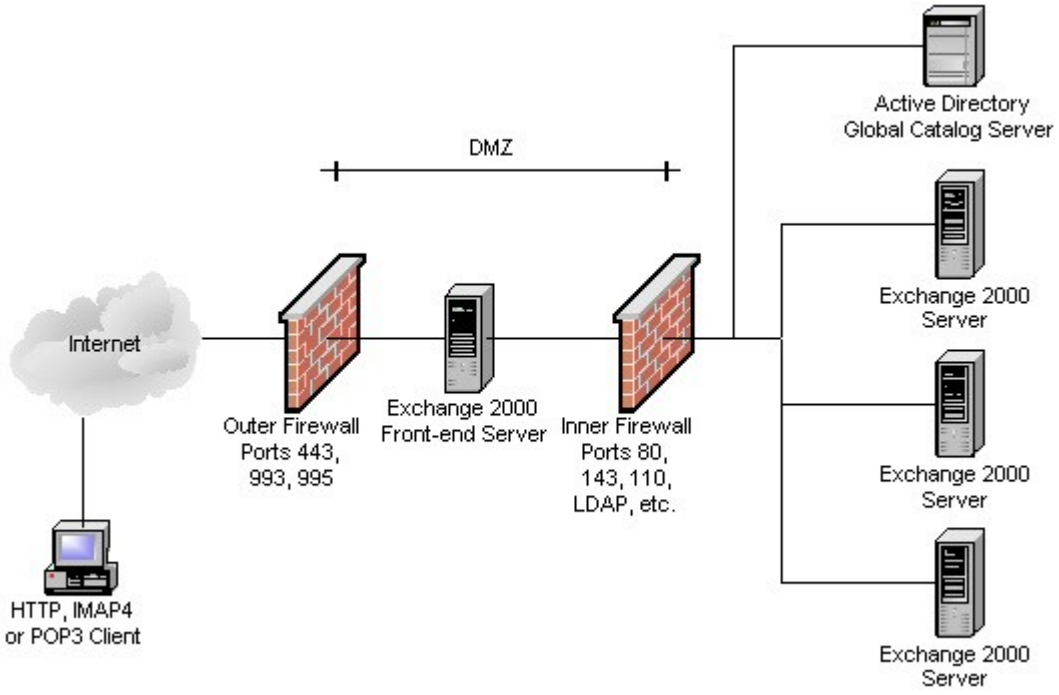

Figure 3: Exchange 2000 Front-end in a DMZ Scenario

Discussion

Typically, corporations that have deployed and standardized the use of a DMZ will have restrictions on the type of network traffic allowed through the Intranet firewall by limiting the network ports that are enabled on the Intranet firewall. However, the front-end server requires certain ports to operate fully.

Issues

Some corporations that have deployed DMZ topologies for other services have policies that restrict computers located within the DMZ from initiating connections with servers inside the corporate intranet. The Exchange 2000 front-end server is not supported in this scenario, because it must initiate connections.

Configuration

Configuring a Front-end Server

A front-end server must be part of the same Exchange 2000 organization as the back-end servers. In fact, it is an ordinary Exchange 2000 server until it is converted to a front-end server. A front-end server must not host any users or public folders.

To set up a front-end server, install the Exchange 2000 server into the organization. Then, using the Exchange System Manager, navigate to the server object, and open the server property page (right-click on the server object, and select **Properties**). Select the checkbox labeled “This is a front-end server,” and close the page. Before the server can be used as a front-end, it must be restarted or HTTP, POP3, and IMAP4 services must be stopped and restarted.

Before restarting the services, dismount and delete the private and public stores on the front-end server. Failing to do so will make the public store inaccessible from the front-end server

Not all Exchange Services are required on a front-end server, depending on the protocols being exposed. The following list shows the Exchange services required for each protocol; all other Exchange Services should be stopped and disabled:

- **HTTP**: No Exchange-specific services required. However, the Windows HTTP service (w3svc) must be running.
- **POP3**: Exchange POP3 (POP3Svc), Exchange Information Store Service (MSExchangeIS), Exchange System Attendant (MSExchangeSA)
- **IMAP4**: Exchange IMAP4 (IMAP4Svc), Exchange Information Store Service (MSExchangeIS), Exchange System Attendant (MSExchangeSA)

To stop and disable Services, use the Services MMC Snap-in.

HTTP-specific configuration

The front-end server's virtual directories and HTTP virtual servers must exactly match those of the back-end. In a default setting, no additional configuration need be done on the front-end – the 'exchange' and 'public' virtual directories are already matching.

POP3- and IMAP4-specific configuration

By default, IMAP4Svc and POP3Svc have a dependency on MSExchangeIS. In practice, however, this is not needed when running as a front-end service. Having the information store process (without any mailbox or public stores mounted) run on the front-end server does not pose any security risk, nor should it affect performance materially. However, it is possible to manually remove this dependency by removing MSExchangeIS from the following registry key:

```
HKEY_LOCAL_MACHINE\
  System\
    CurrentControlSet\
      Services\
        IMAP4SVC | POP3SVC\
          DependOnService
```

Before you edit the registry, make sure you understand how to restore it if a problem occurs. For information about how to do this, view the "Restoring the Registry" Help topic in Regedit.exe or the "Restoring a Registry Key" Help topic in Regedt32.exe. Note that you should back up the registry before you edit it.

If you are running POP3 and IMAP4 in a configuration where you do not have the RPC ports (135, and 1024+) opened between the front-end server and the back-end servers, you must remove these dependencies and must not run the MSExchangeIS service or the MSExchangeSA service on the front-ends. Refer to the Configuration section for additional firewall information.

Configuring a Back-end

Back-end servers must be specially configured to support the front-end server. First, they must be servers in the same domain as the front-end. Users may be on any of the servers in the domain. Public folders can be replicated among

Exchange 2000 servers in any way that is desired. Back-end servers can be accessed directly if required, with no effect on the behavior of the front-end/back-end configuration.

HTTP-specific configuration

Extra virtual directories or virtual servers not available on the front-end server can be created on the back-end servers to support specific applications.

The main configuration that must be done on the back-end server is to use the Exchange System Manager to create a virtual server that maps to each front-end namespace (in the typical environment, all the front-end servers should be mapped to a single name in order to unify the namespace). The new virtual server on the back-end server will be mapped to the single name that represents all the front-end servers. This virtual server will respond to the requests from the front-end server. The virtual server must be configured to match the configuration on the front-end server, including the virtual directories (in particular, 'exchange' and 'public' in the default configuration).

In some circumstances, front-end access to public or private folders might need to be restricted. Remove the 'exchange' (for private) or 'public' virtual directories from the back-end virtual servers to accomplish this.

The specific back-end server configuration steps depend on whether the back-end server is hosted on a cluster or not.

Configuring a non-clustered back-end server

The following steps describe how to create and configure HTTP virtual servers for non-clustered back-end servers.

To create the new virtual server, use the Exchange System Manager. In the HTTP Protocols container for the specific back-end server, add a new virtual server (right-click on **HTTP**, select **New**, **Virtual Server**):

- For a name, Microsoft recommends something of the form "Exchange FE VS (*front-end - hostname*)". Keeping the names of the new virtual servers consistent ensures that the purpose of the virtual server can be easily determined. The name of the virtual server is used only for identification purposes and does not affect its operation.
- Click on the **Advanced** button and add an identity for the virtual server that defines, as a host name, the front-end namespace. This will allow the front-end server to communicate correctly with the back-end server. You may want to add several identities to the virtual server that list all the ways that a user might access the front-end server. For instance, if a front-end server will be used both internally and externally, you may want to list both a short (NetBIOS) name and a fully qualified domain name. See the section below on reducing virtual server creation.
- The TCP port must be port 80, since this is the only port used for communication between front and back-end servers, regardless of the port used by the client to communicate with the front-end server. The SSL port may be left at the default.
- The IP address field may be left as the default "All unassigned" or may be restricted to the particular IP address assigned to the server. Unless

there is a specific reason to restrict the address, the former option is recommended.

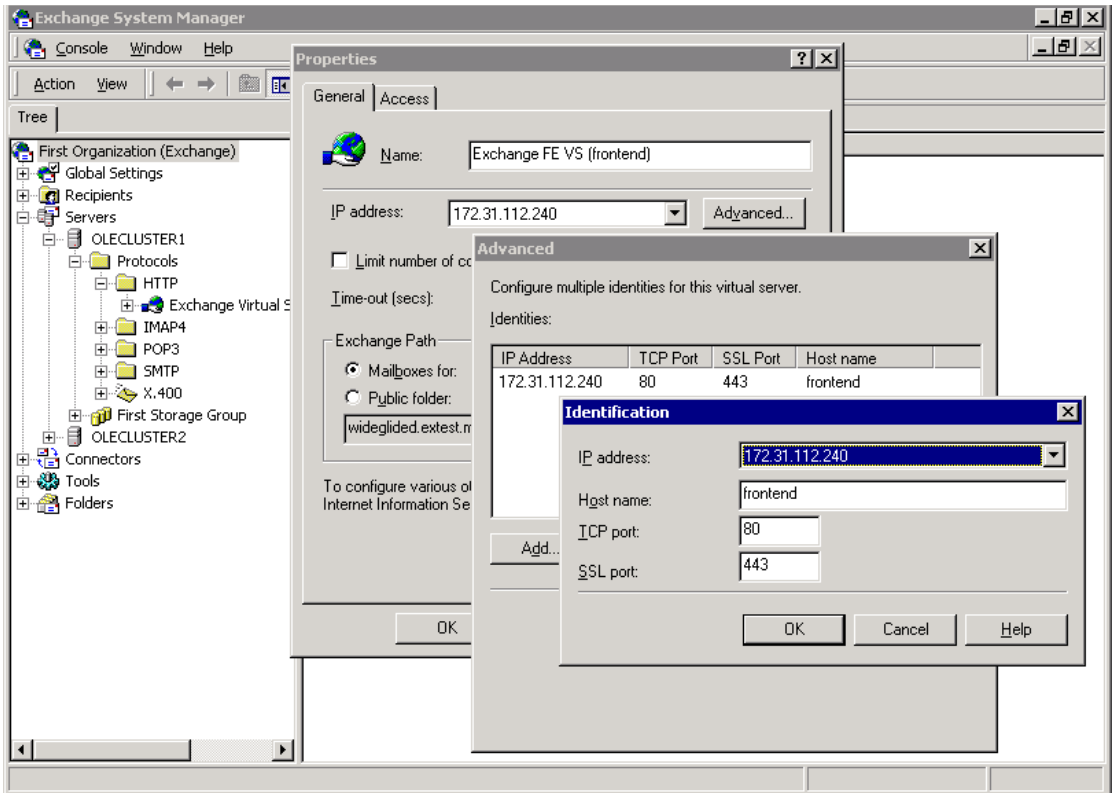

Figure 4: Configuring the virtual server on the backend server

After creating the virtual server, add virtual directories to match those configured on the front-end server. In the default case, an ‘exchange’ and a ‘public’ virtual directory must be created. For any virtual directories pointing at mailboxes, make sure the SMTP domain selected matches the SMTP domain configured on the front-end server. Do not create an ‘exadmin’ virtual directory on the virtual server. This is only used by the Exchange System Manager tool, and is not proxied by the front-end.

To create the new virtual directories, right-click on the new HTTP virtual server in the Exchange System Manager, and then select **New**, **Virtual Directory**.

One Exchange HTTP virtual server must be created for each front-end namespace. Thus, in a hosting environment, where many front-end namespaces (companyA.com, companyB.com, and so forth) are used, there must be a new Exchange virtual server for each of these. This allows for maximum flexibility in determining which resources are available to each hosted company.

Similarly, if a group of front-end servers is being used in a load-balancing scenario, there need not be a virtual server for each physical front-end server, because together, they represent a single namespace.

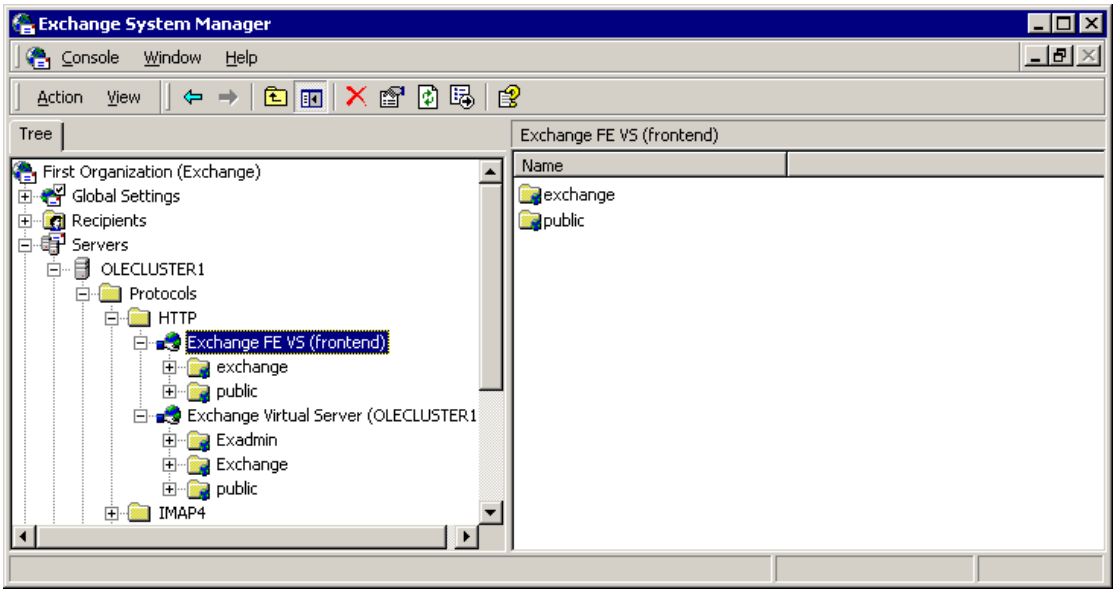

Figure 5: A correctly configured back-end server (assuming a default configuration on the front-end server)

In addition, new virtual servers created on the front-end by the administrator for other reasons (to host web applications, for instance) must be created with the same configuration on the back-end servers (for instance, host names must the same). However, virtual servers that differ from the default virtual server by port number (normal, or SSL) are not supported. All traffic between the front-end and back-end servers is over port 80, regardless of the incoming port used.

Configuring a Clustered Back-end

The following steps describe how to create and configure HTTP virtual servers for clustered back-end servers.

In a cluster environment, Exchange automatically installs HTTP virtual servers in each cluster group. These HTTP virtual servers are configured with the host names of the particular cluster group. To use a front-end server against clustered back-end servers, additional HTTP virtual servers must be created on each back-end server.

On each Exchange server that makes up the clustered environment, use the Exchange System Manager to:

- Create an HTTP virtual server using the guidelines described above for a non-clustered server. Use the naming convention: “Exchange FE VS (*cluster-computername – front-endhostname*).”
- Ensure that unique IP numbers are assigned to each HTTP virtual server according to the IP address that has been assigned to the cluster group. Use the same IP address as that assigned to the automatically-installed Exchange HTTP virtual server. This step is extremely important, because if two HTTP virtual servers share the same configuration, one will be unable to start.
- Create virtual directories as described in the previous section to match those configured on the front-end server.

- Start the HTTP virtual server before continuing, to ensure that it is configured correctly. To do this, right-click on the new virtual server name in the Exchange System Manager, and select “Start.”

Once created, each HTTP virtual server must be added to the cluster group configuration, so that the cluster services can manage it appropriately.

Use the Cluster Administration tool to select each cluster group in the cluster in turn. Using the context menu, create a new cluster resource:

- Name the resource as follows: Exchange FE VS Instance (*cluster-computername – front-endhostname*).
- Set the resource type to “Exchange HTTP Virtual Server Instance”
- When asked, set the virtual server resource to be dependent on the Exchange Information Store resource.

Once created, right-click on the cluster group, and bring the server instance online. Repeat these steps for the other cluster groups in the cluster.

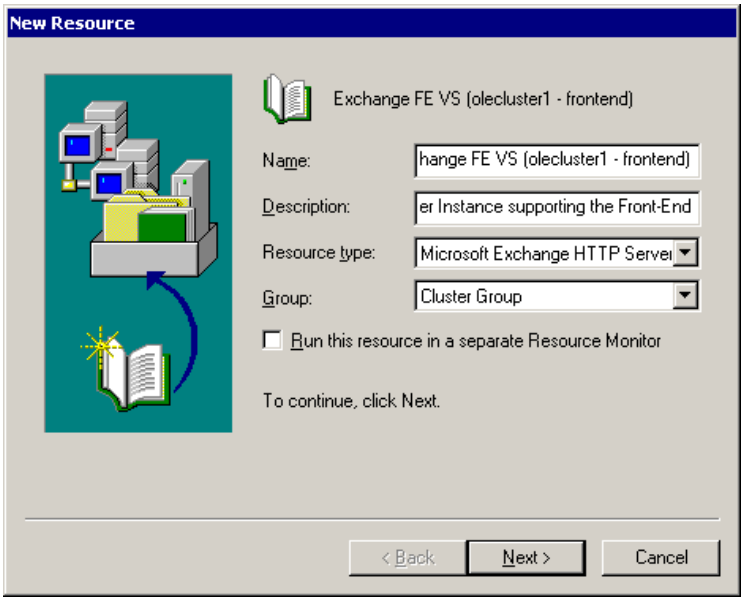

Figure 6: The first step in configuring a new HTTP Virtual Server cluster resource

Once the resources are created and added to the cluster resource, the Windows HTTP service (w3svc) should be restarted on each server in the cluster. This can be done via the command line, the Services snap-in, or by using the IIS administration tool (Internet Services Manager).

Reducing virtual server creation

In some circumstances it may be important to reduce the number of virtual servers created on the back-end servers (for instance, if there are thousands of virtual servers). This should not be done unless the administrator fully understands how HTTP virtual servers work. However, virtual server creation can be reduced using two methods.

First, if each virtual server on the front-end is configured identically, then a single virtual server can be created on each back-end server with several host names associated with it. In addition, virtual servers only need to be created on back-end servers that may be directed to by the front-end server. For instance, if a front-end server contains virtual servers for companyA.com, and for companyB.com, and all companyA.com users are hosted on server A, then on server A, only a virtual server for companyA.com is required. If configurations change, then changes may need to be made to the back-end server. An analysis of the users and data on a server can be made to determine if users will ever be directed to a particular back-end server. For instance, on a server that has no public store, the 'public' virtual directory is not necessary.

Configuring Firewalls

This section discusses the configuration of firewalls for use with an Exchange front-end / backend topology, as well as any additional configuration required on the front-end servers in these environments.

Configuring an Internet firewall

In Internet scenarios, a firewall is required to be set up between the corporation and the Internet. The firewall must be configured to allow requests to certain IP addresses and over certain TCP and UDP ports. See the following list for the ports required for different services.

Port Number/Transport	Protocol
443/TCP	HTTPS (SSL-secured HTTP)
993/TCP	IMAPS (SSL-secured IMAP)
995/TCP	POP3S (SSL-secured POP3)
25/TCP	SMTP

Table 1: Ports required to be open on the Internet firewall

Configuring an Intranet firewall

When setting up a Demilitarized Zone (DMZ), a firewall must be configured between the front-end server and the corporate Intranet.

Basic protocols

All of the supported protocol ports need to be open on the inner firewall in all cases. The SSL ports do not need to be open, because SSL will not be used in communication between the front-end and the back-end servers. These ports are mandatory.

Port Number/Transport	Protocol
80/TCP	HTTP
143/TCP	IMAP4
110/TCP	POP3
25/TCP	SMTP

Table 2: Protocols ports required for the Intranet firewall

Active Directory Communication

In order to communicate with Active Directory, the Exchange front-end server requires (Lightweight Directory Access Protocol) LDAP ports to be open. Windows

2000 Kerberos authentication is used; therefore, the Kerberos ports must also be open. These ports are required.

Port Number/Transport	Protocol
389/TCP	LDAP to Directory Service
3268/TCP	LDAP to Global Catalog Server
88/TCP	Kerberos authentication
88/UDP	

Table 3: Ports required for Active Directory Communication and Kerberos

There are two sets of optional ports that can be opened in the firewall. The decision as to whether to open them depends on the policies of the corporation. Each decision provides a tradeoff between security, ease of administration, and functionality. First, the easy choice:

Domain Name Service

In order for server names to be looked up correctly (for instance, converted from names to IP addresses), the front-end server needs access to either a Domain Name Service (DNS) server or a hosts file that provides the mapping. In a default Windows 2000 domain, Active Directory servers function as DNS servers, so these are the servers that should be made accessible through the firewall over the DNS ports.

Port Number/Transport	Protocol
53/TCP	Domain Name Service (DNS) Lookup
53/UDP	

Table 4: Ports required for access to DNS Server. Use hosts file if these ports aren't open.

How to configure if DNS ports are not open: If these ports are not open in the inner firewall, then a hosts file must be created for each mapping. The Windows 2000 TCP/IP stack will use the following file: %SystemRoot%\System32\drivers\etc\hosts. Edit this file to create a name-to-IP-address mapping for every server that the front-end server might contact. This generally includes every Exchange 2000 server in the organization, and each Active Directory or Global Catalog server that the front-end server might contact. Instructions for how to create the mappings are in the default hosts file. The most important thing about using a hosts file is to ensure that it is kept up-to-date when changes are made to the organization.

Service Discovery and Authentication

Windows 2000 uses Remote Procedure Call (RPC) calls to perform Active Directory service discovery and to perform client authentication. In order to enable these features, RPC ports must be opened. See Table 5 for the list of required ports.

Port Number/Transport	Protocol
135/TCP	RPC port endpoint mapper
1024+/TCP	RPC service ports
445/TCP	Netlogon

Table 5: RPC ports needed for Service discovery and authentication.

How to configure if RPC ports are not open: Corporations that have standardized on DMZs at their firewalls will often have restrictions on the type of traffic that

can be sent from within the DMZ into the corporate Intranet. If open RPC ports are not allowed between the DMZ and the corporate Intranet, certain front-end server features must be disabled or specially configured. First, without RPC access to the Active Directory servers, the front-end server cannot authenticate clients. This means that features that require authentication do not work correctly, including implicit logon and public folder tree access. Public folder access is possible, but without being able to determine the identity of the accessing user, no load-balancing is possible. When configuring a front-end server in this environment, disable authentication and enable anonymous access.

In addition, because Windows 2000 is unable to locate its Domain Controllers and Global Catalog Servers without RPC service discovery, the Exchange front-end server must be pre-configured with the server names of these servers and their back-ups. A registry file supplied with this document contains some registry keys will be checked by the front-end server if it is unable to locate the servers via RPC. Edit the file appropriate to the environment and import it into the registry on the front-end server.

Before you edit the registry, make sure you understand how to restore it if a problem occurs. For information about how to do this, view the "Restoring the Registry" Help topic in Regedit.exe or the "Restoring a Registry Key" Help topic in Regedt32.exe. Note that you should back up the registry before you edit it.

As with the hosts file described above, the most important thing is to keep the contents of these registry entries up-to-date as the domain evolves.

If you are running IMAP4 or POP3 on the front-end servers, with the above RPC ports closed, you must ensure that the MSExchangeIS and MSExchangeSA services is not running on the front-end server. Refer to the “Configuring a Front-end Server” section for more information on how to remove IMAP4’s and POP3’s dependency on the MSExchangeIS service.

Additional Configuration

Securing Communication – Client to Front-end server

In order to secure data transmitted between the client and the front-end server, Microsoft recommends that the front-end server be SSL-enabled. In addition, to ensure that users’ data is always secure, access to the front-end server without SSL should be disabled (this is an option in the SSL configuration). It is of critical importance to protect the network traffic by using SSL when using Basic authentication in order to protect the users’ passwords from network packet sniffing.

To set SSL up for use with HTTP, use the Internet Services Manager and follow the IIS documentation instructions to request and install an SSL certificate on the default Web site, or whichever virtual server needs protection.

SSL need not be configured on the back-end servers when using a front-end server, because the front-end server does not support using SSL to communicate to the back-end servers.

Troubleshooting tip: Ensure that the Windows 2000 License Logging Service is running on the front-end server. IIS will not allow more than ten simultaneous SSL connections unless this service is running.

Securing Communication – Front-end to Back-end

HTTP communication between the front-end and the back-end servers is not encrypted. In many cases, for instance, when the front-end and back-end servers are maintained in a separate subnet, this is not a concern. However, if front-end and back-end servers are kept in separate subnets, and network traffic must pass over unsecured areas of the corporation, it may be necessary to encrypt this traffic to protect passwords and data.

Windows 2000 supports IP Security (IPSec), an Internet standard that enables two Windows 2000 servers to encrypt any traffic between them at the IP (Internet Protocol) layer. This protocol can be used to secure the traffic between the front-end and the back-end servers. This kind of encryption will have a performance hit on both the front-end and the back-end servers. The precise extent of the performance hit depends greatly on the type of encryption used.

Enabling IPSec should only be done with a thorough understanding of the workings of the protocol and its administration. The following steps should provide a rough guideline of what needs to be done.

The Windows 2000 IPSec feature allows a server to encrypt any and all traffic. However, in this scenario, it is best to limit the encryption to only the HTTP traffic. The IPSec policy file included with this document shows an example of how to do this. Filter the IPSec policy to apply only to traffic initiated by the front-end server that is being sent to port 80 on a remote server. To examine the included policy file, load up the IP Security Policy Management snap-in using the Microsoft Management Console (MMC), and import the policy. Use the snap-in to assign it for use on the front-end server.

Do not use the included policy on the back-end servers. IPSec should be configured on the back-end servers so that they respond appropriately if a request for IPSec communication is received. However, they should not require that all communication from all clients be encrypted using IPSec. Windows 2000 has three included IPSec policy files. One of these, labeled “Client (respond only),” will have the desired effect. With this policy enabled on the back-end, the front-end will be able to use IPSec to communicate safely with the backend, while other clients (including MAPI clients like Outlook 2000) and servers will be able to communicate with the back-end server without needing to use IPSec.

When using IPSec in a DMZ environment, changes must be made on the inner (or Intranet) firewall to support it. First, HTTP (port 80/TCP) is no longer required, and should be blocked. Second, the IPSec negotiation port at 500/UDP is required. IPSec also requires IP id 50 and 51 to be open on the inner firewall.

Note IPSec encryption occurs after the application (in this case, Exchange 2000) has passed the request to Windows to send to the server. So, as far as Exchange 2000 is concerned, the request was made over HTTP using TCP port 80. However, before the traffic leaves the server, it is intercepted, encrypted and sent over a separate channel (IP ids 50 and 51). Thus, the encryption is

transparent to the Exchange 2000 applications running on each server, and the fact that the data never used port 80 is not an issue to these applications.

Front-end/Back-end Topology: Checklist

- ☑ Install the front-end server
  - Remove public and private stores from the server.
  - Stop and disable all unnecessary services.
  - *Optional:* Enable SSL encryption.
  - *Optional:* Assign front-end server IPSec Policy.
- ☑ Configure the back-end servers
  - Create virtual servers to match the front-end server.
  - Create virtual directories to match those on the front-end server.
  - *Optional:* Assign back-end server IPSec policy.

If placing on the Internet:

- ☑ Configure the front-end server
  - *Optional:* Set up hosts file (if DNS ports are closed).
  - *Optional:* Set up directory access registry entries (if RPC ports are closed).
  - *Optional:* Switch to anonymous access (if RPC ports are closed).
- ☑ Configure firewalls
  - Enable access to mandatory protocol ports.
  - *Optional:* Enable access to additional ports.

Front-end/Back-end Topology: Troubleshooting steps

The following steps should be followed when troubleshooting reported or observed problems with a deployed front-end / back-end topology.

The first step should always be to attempt to connect directly to the user's back-end server using the protocol that does not appear to be functioning correctly through the front-end server. If connecting to the back-end does not work as expected, or displays the same problems that the front-end server displays, then follow standard trouble-shooting steps for Exchange 2000 topologies.

If the back-end server is functioning as expected, then the second step should be to ensure that the front-end server is able to successfully connect to (a) the global catalog servers and, (b) the back-end server in question. This is particularly important when the front-end server is deployed in a DMZ scenario (in particular, check the hosts file and registry keys if they are in use).

To test whether a server can be reached from the front-end, use the Windows 2000 telnet.exe application on the front-end server to open connections from the front-end to the back-end server over the protocol ports and the LDAP port. For example, to test that HTTP connectivity from the front-end to the back-end is working, run the following command on the front-end server:

```
telnet backendserver 80
```

If no error appears, then the connection is open (in the case of HTTP, press Enter twice to close the connection if a blank screen appears).

To test whether a server name is correctly resolved to an IP address, use the nslookup.exe tool to test DNS resolution.

Problems experienced with front-end / back-end deployments can often be traced to the inability of network traffic to flow from the front-end server to the correct servers due to misconfigurations on the server or the network routers. In all cases, Event Log entries may help troubleshoot the particular issue.

HTTP-specific troubleshooting tips

The previous steps are useful for any protocol. However, some of the error messages that are returned when using HTTP through the front-end server can help target investigations. Depending on the configuration of your browser, the error number may be obvious, or may be somewhere in the text of the response.

401 Unauthorized: This error means that the user entered an incorrect username and/or password. Remember that authentication requires a username of the format *domain**password*.

404 File Not Found: This error means that the URL entered was not valid. Look for typographical errors in the name of the user or other parts of the URL. If there are none, check the configuration of the HTTP virtual server on the back-end server, checking for typographical errors in the names of virtual directories that were created.

500 Internal Server Error: This error occurs most often in a locked-down DMZ scenario, and is a symptom of the front-end server's inability to connect to domain controllers through a closed-down Intranet firewall. Check the hosts file and registry key settings carefully against the actual configuration of the organization.

503 Service Unavailable: This error is generated in two main situations. First, if the Information Store service on a back-end server is stopped, or the user's database is dismounted, then the back-end server itself will return this error. The front-end server will generate this error if it is unable to contact a valid back-end server. This could be because the back-end server is down for maintenance, or because of network issues between the front-end and back-end servers. Use telnet.exe to verify that the front-end server can connect to the back-end servers.

Acknowledgements

The author would like to thank all the people whose hard work made this document possible, including Becky Anderson, Karim Batthish, Brian Deen, Mike Gahrns, Mindaugas Krapauskas, Chris Simison, Joel Soderberg, and Anuradha Rawal.

For more information

- Internet Standard Documents
  - HTTP: RFC 2616 (www.ietf.org/rfc/rfc2616.txt)
  - POP3: RFC 1939 (www.ietf.org/rfc/rfc1939.txt)
  - IMAP4: RFC 2060 (www.ietf.org/rfc/rfc2060.txt)
    - IMAP Referrals: RFC 2221 and RFC 2193

- IPSec: IP Security for Windows 2000 Server (http://www.microsoft.com/WINDOWS2000/library/howitworks/security/ip_security.asp)

Microsoft Exchange Internet Site: http://www.microsoft.com/exchange/

◆ ◆ ◆ ◆